

Project Specification - Group 3

1. Project Description

We are building a first-person single-player PvP shooter, where players navigate a maze in a hunt for their fellow contestants. The game can be played with 2 or more players. Once a player is killed, they are eliminated from the game. The game ends when only one player remains, and that last player is the winner.

While the PvP type game is a common one, the maze aspect of the game opens up many creative possibilities. Players can gain advantages in the form of special ability power-ups lying throughout the maze. When a player picks up a power-up, they are granted a one-time use special ability. These special abilities include temporarily being able to see the locations of other players in a birds-eye view, phasing through walls, shattering a wall, and resetting the entire arena. These power-ups provide significant advantages to players, and will provide incentives for players to move actively resulting in exciting action.

Being able to find opponents will be an exciting aspect of the game. We are considering adding noises like breathing and footsteps that grow louder as an opponent's approach nearer. We are also considering footsteps leaving a trail with a unique color corresponding to every player, allowing players to be stalked or perhaps set up an ambush.

To prevent long stalemates with players parked in distant corners, we are considering adding a feature where walls begin disappearing one by one after a certain amount of time passes.

Features:

Must Haves:	<ul style="list-style-type: none">• Maze• Player spawn/movement• Shooting/Killing Mechanic• Special Ability Powerups
Would Be Really Nice:	<ul style="list-style-type: none">• Automatically generated mazes for variety• Multiple rounds featuring different types of mazes (Ex: weapons cornucopia in center, multi-level maze) and concepts (unique themes, designed minimap)

	<ul style="list-style-type: none"> • Player trails • Disappearing walls to force an endgame • Increasingly loud footsteps and breathing as players grow closer
If ahead of schedule:	<ul style="list-style-type: none"> • Damage to maze after shooting, blood trails after a kill • Team v Team mode • Third Person viewing option • Maze traps

2. Group Management

In terms of group management, no one particular person will be charged with a role overseeing the development of the project. Instead we will hold each other accountable and check in with each other through consistent communication. Development roles are listed in the section below.

Decisions will be made by consensus from discussion. We will primarily be communicating through Discord and Zoom meetings. To ensure we are working efficiently together, we will use Notion, an online planning tool, to update our progress on goals and milestones together.

We will remain in close communication, so we will be aware if we veer off schedule. If this happens will shift effort and attention to the most challenging issues, so that we can get back on track. It is important we understand that obstacles will reveal themselves as we take on this challenging task and not get too frustrated by setbacks.

Weekly status reports will be sent to and compiled by Robert and Alan, who will then publish the reports. They will be responsible for ensuring the reports are sent in on time.

3. Project Development

We will shift around roles as the needs of the project evolve, but these will be the initial roles.

Art: Alan Tran

Audio: Alan Tran

Graphics: Lucas Hwang

Gameplay: Mitchell Zhang, Kabir Nagrecha, Bowen Shi

Networking: Robert Screven, Michael Than

Website: Robert Screven, Alan Tran

For graphics, we will be using OpenGL. We are considering using OpenGL to design characters and objects. Based project feedback provided on the course website, we are also considering using Boost, a C++ library, for networking purposes. Our game systems will likely be programmed in C or C++. Catch and GoogleTest will likely be used for C++ unit testing and mocking. Visual Studio Code will be among the IDE's used.

A combination of unit testing, integration testing, and user testing will be used to ensure a well-running game.

Code will be well-commented so it is easier to collaborate and diagnose potential issues. External documentation will be provided for weekly deliverables, possibly with the help of code-parsing documentation tools.

4. Project Schedule

High-Level:

1. Basic 3D Graphics Implementation - Graphics engine is running such that an object can exist within it and the object and camera can move. Target: End of Week 2.
2. Echo network - Server can echo a message back to the client. Target: End of Week 2.
3. Graphic-Network Integration - Client sends input to server, the server receives this message, server updates world state, server sends updated world state to client, client sees world state updated. Target: End of Week 3:
4. Basic Design - Basic playable character designs completed and a generic maze is finished. Target: End of Week 3.
5. Player-World Gameplay - Players input communicated through the network protocols moves their character around the generic maze, object collision with walls.
6. Player-Player Gameplay - All players can interact with each other (through shooting). A player that is shot too often will die. Target: End of Week 6.
7. Player-World Gameplay Depth - Power-ups are added to the maze. When players pick up a power-up they are granted one-time use of the power-up's ability. Target: End of Week 7.
8. Textures and Sounds - Life is given to the game with more complex character designs, the addition of sound design such as footsteps and breathing, and player trails. Target: End of Week 7.

9. Design Freeze - The game has progressed to a complete vision. Only minor additions will be made to the game after this point. Target: End of Week 9.

Low-Level:

Week	Graphics	Networking	Gameplay	Art/Audio	Other
1	Render 3d world and create an object.	Research available libraries	Brainstorm game design	Visualize character design, theme	Start website, post project spec
2	Move object based on input, get player camera working, research how to integrate with networking library	Select design choice, create echo network.	Decide what objects must exist and what value fields they must hold, begin implementation with dummy network methods	Sketches of character designs	
3	Integrate graphics and network	Integrate graphics and network		Rough character design models, generic maze	Beautify website
4	Load in player and maze models	Design messages for required gameplay values	Hook up player-world gameplay mechanics (player movement and environment collisions)		
5	Work on character animations		Two player interactions (shooting)	Basic power-up ability visuals(bir	

				d-eye view)	
6	Optimize collisions and work on minimap	Decide what's needed for client-side rendering	Multiple player interactions, scaling	Sound effects	
7	Fix graphics issues appearing after full gameplay integration		Complex maze interactions (power-up abilities)	Player trails, splatter	
8			Automatic maze generation		
9				Final Design Touches	
10	Fix bugs and prepare for final demo	Fix bugs and prepare for final demo	Fix bugs and prepare for final demo		